* **ME-5103 IN-CIRCUIT EMULATOR**

ME-5103

单片机仿真开发系统

用户手册

User Manual

万利电子有限公司

Manley Electronics Co., Ltd. makes no warranty of any kind with regard to this document, including, but not limited to, the impied warranties of merchantibility and fitness for a particular purpose. Information in this document is subject to change without notice and does not represent a commitment on the part of Manley Electronics Ltd. No part of this manual may be reproduced or transmitted in any form or by any means, electronical or mechanical, including photocopying or recording, for any purpose, without the express written consent of Manley Electronics Co., Ltd.

Copyright © 1995 Manley Electronics Co., Ltd.

目录

引言 1
ME-5103单片机开发系统简述 1
ME-5103单片机在线仿真器 2
MBUG调试器 3
ME-5103选件 4
第一部分 ME-5103仿真器 5
安装与连接 5
安装ME-5103 5
检查跳线器 5
连接电源 5
连接仿真电缆 6
连接RS-232电缆 6
上电测试 6
ME-5103 键盘功能 6
RESET 系统硬件复位 8
编程与调试命令 8
MON 监控键盘 8
0-F 数字键 8
NEXT,LAST 地址增量和减量键 10
TRACE 跟踪键 13
STEP 单步键 13
HIST 历史键 13
BP,BT 断点和断点次数键 14
EXEC 执行程序键 15
MAP 仿真模式和写模式键 17
数据块命令 19
BTOP,BEND 设置块首和块末键 19
COMP 数据比较键 20
MOVE 数据块移动键 21
FFIS,FFNO 搜寻匹配与不匹配数据键 22
系统实用命令 23
OFFSET 偏移量计算键 23
HEX,DEC,+,- 十六进制和十进制转换, 加减法运算键 24

FUNC 用户功能键 25
USER 用户命令键 25
ME-5103仿真器的硬件结构 26
ME-5103仿真器的地址空间和存储器扩展 26
ME-5103用户数据区 27
ME-5103显示字符表首地址(CE80H-CE81H) 28
ME-5103键值表首地址(CE82H-CE83H) 28
用户命令程序入口表地址(CE84H-CE85H) 28
用户功能程序入口表地址(CE86H-CE87H) 28
8个LED显示数据的地址(CE88H) 28
数码管显示数据的地址(CE89H) 29
ME-5103单板机 29
中断与定时器的处理 29
中断服务优先级寄存单元ISP 30
中断服务优先级寄存单元ISP的使用规则 30
ME-5103关于中断指令的处理 31
ME-5103对定时器的控制 31
第二部分 MBUG调试器 33
安装和运行MBUG 33
MBUG窗口环境 34
常用功能键 35
MBUG主菜单命令 35
HELP, ? 帮助 36
OS Shell DOS 功能调用 36
Load 装入调试文件 36
Setup 设置寄存器窗口 38
Assemble 交叉汇编器 38
Disassemble 反汇编器 38
Run 运行 38
Quit 退出 MBUG 39
调试运行 39
调试运行命令 40
F2 BP, Ctrl-B 设置断点 / Ctrl-T, BT 设置断点次数 40
F3 PC, Ctrl-N 设置程序计数器 41
F4 Here 运行程序到光带处 41
F5 Step, Ctrl-F5 单步, 自动单步 41
F6 Zoom 改变窗口尺寸大小 42

F7 Symbol 符号开关 42
Ctrl-F7 激活符号表窗口 42
F8 Trace, Ctrl-F8 跟踪, 自动跟踪 42
F9 Exec 全速运行. 42
F10 Menu 反回主菜单 42
数据块操作命令 43
Shift-F2 Comp 块比较 43
Shift-F3 Move 块移动. 43
Shift-F4 Fill 块填充. 43
Shift-F5 Match 寻找相同字节 43
Shift-F6 Unmatch 寻找不同字节. 43
Shift-F7 Save 保存程序代码到磁盘文件. 43
行汇编功能 44
第三部分 MBUG汇编器 45
MBUG汇编器的字符 45
MBUG汇编器的符号 45
MBUG汇编器的常数 45
MBUG汇编器对文件的限制. 45
MBUG汇编器的运算符+,-,HIGH,LOW 46
MBUG汇编器的伪指令 46
DB 46
DW 47
DS 48
EQU 48
XDATA 48
DATA 49
BIT 49
.8XXXX 50
ORG 50
END 50
错误信息 51
附录A ME-5103仿真器模式 52
附录B ME-5103仿真器调线 53
JP1 晶体振荡器内/外部选择 53
JP2 MCU片内串行口TTL-RS232电平转换选择 53

JP3 用户系统电源供电选择 53
JP4 U4(0A000H-0BFFFH)用户写允许选择. 53
JP5 仿真器/单板机选择 54
附录C ME-5103仿真器连接器引脚定义 56
附录D ME-5103通讯电缆连接. 57
附录E ME-5103监控子程序. 58

引言

ME-5103单片机开发系统简述

ME-5103单片机开发系统是万利电子有限公司设计生产，用于对8051系列单片机软件、硬件调试的高性能普及型开发工具。ME-5103单片机开发系统包括：ME-5103 单片机在线仿真器、MBUG调试器及交叉汇编器。ME-5103单片机开发系统简单、易学，为学习、使用、开发8051系列单片机提供了一个廉价的开发工具和新的工业标准。ME-5103单片机在线仿真器自带键盘和显示器能够单独运行，配合MBUG 调试器运行时，仿真调试更加直观方便。

使用ME-5103单片机在线仿真器和MBUG软件调试器及交叉汇编器， 能够对用户的源程序进行汇编，装入目标代码到 ME-5103仿真器或目标系统， 仿真用户目标系统的软、硬件，具有跟踪、单步、断点、全速运行多种命令，观察和修改数据、文本调试、汇编、反汇编等多种功能。

ME-5103单片机开发系统用户手册详细描述ME-5103单片机在线仿真器、MBUG 软件调试器和交叉汇编器的安装、连接、使用。

ME-5103单片机开发系统支持开发下列器件：

8051,80C51,8052,80C52(EA=0)
8031,80C31,8032,80C32

附加仿真适配器选件支持开发下列器件:

INTEL	8751,87C51（EA=1）
PHILIPS	80C451,80C552,80C652, 87C750,87C751,87C752
ATMEL	89C51（EA=0/1），89C2051

ME-5103产品和基本附件

完整的ME-5103单片机开发系统包含:

1. 带有8K RAM的ME-5103单片机在线仿真器
2. 仿真电缆
3. 装有MBUG调试器的软盘
4. RS232电缆
5. 用户手册

ME-5103单片机在线仿真器

ME-5103单片机在线仿真器的仿真形式为CPU仿真。对目标系统仿真时，必须移去目标系统上的MCU(单片机)芯片，将40线仿真电缆的DP40仿真头插入目标系统的MCU插座，另一头与ME-5103 仿真器的 CN1相连，RS232电缆两端分别与 IBM PC 机的串行口(COM1或COM2)，ME-5103 仿真器的CN6连接。提供 5V直流电源至ME-5103仿真器标有5V“字样的焊盘或从与 IBM PC连接的RS232电缆中引出。

ME-5103 仿真器提供了多达48K的仿真存贮器空间，用于仿真用户程序或数据。地址 0000H-1FFFH 的 8K 空间具有后备电池保护功能。2000H-9FFFH (U5)，A000H-BFFFH (U4)用于扩展仿真存贮器,可以插入RAM,EPROM,EEPROM。监控程序使用ME-5103仿真器C000H-DFFFH的程序和数据空间，两个26pin IDC插座用于扩展ME-5103仿真器未使用的存贮器空间。

31键的键盘由一个复位键、16个数字键、14个功能键组成。6个七段LED数码管用于显示地址和数据，数码管上方的8个LED用于显示用户定义寄存器的内容， 数码管左侧的 2个LED显示当前的仿真模式。

使用ME-5103仿真器的键盘和显示器，能进行跟踪、单步、设置断点及次数、全速运行程序，检查修改片内RAM、SFR、仿真存贮器

和目标系统存贮器，数据块比较、 移动、填充，搜寻数据块内匹配和不匹配的字节， 显示指令运行的历史路径，计算相对转移的偏移量等操作。

ME-5103 仿真器本机晶体振荡器频率为6MHz，使用外部晶体振荡器的频率范围为4-16MHz。通讯由监控程序软件通过8155 I/O口两个位构成的串行口与的与 IBM PC的串行口实现， 自适应波特率。MCU上的TXD，RXD可通过跳线配置到9芯D型插座上。

ME-5103仿真器能够简单容易地通过跳线作为单板机使用。

MBUG调试器

MBUG调试器是一个基于DOS的应用程序，通过串行口对ME-5103 进行控制和操作。 MBUG 调试器对汇编以及高级语言的调试提供了方便、灵活、有效的窗口界面。

MBUG的窗口界面分为存贮器、寄存器窗口，用于显示和修改寄存器、特殊功能寄存器、外部数据、代码、位寄存器、变量的内容；工作窗口，用于源文件、标号、反汇编的方式的调试程序。数据块操作，具有单步、跟踪、断点及全速运行、行汇编和数据块操作等操作命令；主菜单包含装入程序、交叉汇编、反汇编等多种功能。

MBUG调试器内含交叉汇编器，由主菜单或热键激活。用户程序经汇编后产生符合 INTEL HEX 标准格式的文件。这个文件能被ME-5103装入，在源文件文本方式下调试。

MBUG调试器的运行环境：

1．IBM PC及兼容机286，386，486
2．MS-DOS 3.1以上版本
3．300K以上的有效内存
4．一个RS-232电平的串行口

ME-5103选件

单片机仿真适配器

ME-8751	仿真8751,87C51,89C51 (EA=1)
ME-451	仿真PHILIPS 80C451
ME-552	仿真PHILIPS 80C552
ME-752	仿真PHILIPS 87C752,87C751,87C750
ME-89C251	仿真ATMEL 89C2051

编程器选件

MP-100

EPROM
2764/128/256/512/010/020/040/080
EEPROM
2817/64/256/010/020
FLASH MEM
29C256/512/010/020
MCU
87C51/52/54/58，89C51/52/55
89C2051/1051
PLD
AT16V8/20V8/22V10

第一部分 ME-5103仿真器

安装与连接

该部分详细叙述ME-5103单片机在线仿真器电源、跳线、通讯电缆、仿真电缆的安装、连接方法和键盘功能。在安装前，请检查ME-5103包装内的随机附件是否齐全，然后移去 ME-5103 的塑料盖进行下面的安装。

检查跳线器

在ME-5013仿真器PCB板上有5个供用户配置的跳线: JP1, JP2, JP3, JP4, JP5, 在出厂时这5个跳线器的跳线位置见图1-1.其作用是:

JP1　晶体振荡器内/外部选择
JP2　MCU片内串行口TTL-RS232电平转换选择
JP3　用户系统电源供电选择
JP4　U4(A000H-BFFFH)用户写允许选择
JP5　仿真器/单板机选择

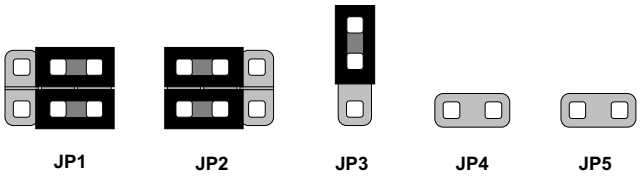


图1-1 ME-5103 出厂调线器设置

连接电源

ME-5103 仿真器采用单5V电源供电，要求电源容量在1.5A以上。连接电源有以下两种方式：将电源线直接焊至PCB板的左侧标有"5V"字样的焊盘或从通讯电缆的9芯D型插头中引出(见附录D)。

警告：任何时刻都不得将电源反接或将高电压加入.

连接仿真电缆

ME-5103仿真器在对目标系统仿真时，须将目标系统的 MCU移去，然后将40线仿真电缆的一头(IDC40)插到ME-5103仿真器的CN1，另一头 (DIP40)插到目标系统的 MCU的插座上。连接方向按ME-5103仿真器的地线与目标系统的地线相连为准。

注意：仿真MCU的引脚与仿真头引脚名一一对应，但并不全部连通。

警告：ME-5103仿真器的地线与目标系统的地线必须连通, 否则将有可能损坏ME-5103仿真器或目标系统。

连接RS232电缆

ME-5103仿真器与 IBM PC连接时，将随机配置的RS232通讯电缆9芯D型插头连至 ME-5103 仿真器的 CN6，25芯 D型插头连至IBM PC的串行口COM1或COM2，如 IBM PC的串行口是9芯插座。请另购9芯转25芯转换器或按附录D 将RS232通讯电缆的25芯插头改为9芯插头。

上电测试

在完成以上安装连接后，必须仔细检查连接是否正确，电源是否满足要求。然后再打开电源。此时ME-5103仿真器的数码管上将显示"P"。如没有显示，须立即关闭电源。

ME-5103 仿真器在接通电源后，即能独立工作。如须在 IBM PC上进行操作，在提示符下键入MBUG，选择调试运行 (RUN)项，屏幕上将显示当前ME-5103仿真器的各种内容和状态。通讯如不正常,按Ctrl-C键选择另一串行口再试。

ME-5103键盘功能

图1-2是ME-5103键盘布置图，左侧16个为十六进制数字键,右侧

为14个功能键和一个硬件复位键。16个数字键也可作为当前寄存器区R0-R7和常用16个SFR的直接访问键。

表1-1显示了14个功能键和硬件复位键的定义。它们的使用方法及应用举例见"ME-5103的使用"部分。

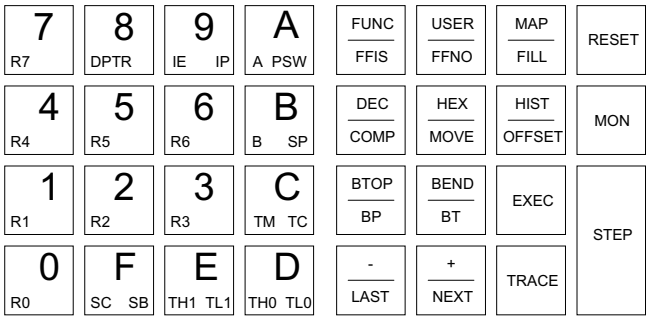


图**1-2 ME-5103 键盘布置图**

表1-1 功能键和硬件复位键

RESET	系统复位键
MON	监控键
LAST/-	上一地址/减键
NEXT/+	下一地址/加键
TRACE	跟踪键
STEP	单步键
EXEC	执行键
BP/BTOP	断点/块首键
BT/BEND	断点次数/块末键
COMP/DEC	数据块比较/十进制键
MOVE/HEX	数据块移动/十六进制键
OFFSET/HIST	偏移量计算/历史键
FILL/MAP	填充/仿真区域键

FFIS/FUNC	搜寻配匹/用户功能键
FFNO/USER	搜寻不配匹/用户命令键

RESET系统硬件复位

按RESET键使系统硬件复位。此时程序计数器PC和所有的SFR初始化为上电状态。不改变 ME-5103 的仿真模式、状态和片内RAM的内容, 7段数码管显示"P"，等待接收命令。

上电复位强制使 ME-5103仿真器的仿真模式为0，七段数码管上方的8个LED设置成显示累加器 A的内容，其它与按 RESET键使系统硬件复位相同。

短路跳线器JP5，将ME-5103仿真器作为单板机使用时。无论上电复位或按RESET键使系统硬件复位,程序均从U6（地址0000H-1FFFH）开始运行用户程序。

编程与调试命令

MON -监控键

在监控程序有效管理 ME-5103的任何时刻，按下MON键都在七段数码管上显示"P"，等待接收命令。MON 键通常用于终止一个命令，执行一个新的操作。如须终止正在运行的用户程序，只有使用RESET键，但所有的SFR将改变为上电状态。

0-F -数字键

键盘的左半部分为16个数字和寄存器地址键，用于输入地址和数据。当数字键与NEXT和LAST键使用输入地址时，能够访问寄存器、SFR和存贮器。根据输入地址的位数，分别位:

一位数字地址

表2-1表示一位数字地址所访问当前寄存器区的寄存器和SFR对照表。

表2-1　一位数字地址与寄存器和SFR对照表.

数字地址	寄存器	数字地址	寄　存　器
0	R0	8	DPTR
1	R1	9	IE　IP
2	R2	A	A　PSW
3	R3	B	B　SP
4	R4	C	TMOD TCON
5	R5	D	TIMER0
6	R6	E	TIMER1
7	R7	F	SCON SBUF

二位和三位数字地址

二位和三位数字地址范围在00H-7FH或000H-07FH时，访问低128字节的片内 RAM；二位数字地址范围在 80H-FFH 时，访问高128字节的片内RAM (此时对片内有256字节RAM的MCU有效，如8052)；三位数字地址范围在080H-0FFH时，访问SFR地址空间。见表2—2。

表2-2 二位和三位数字地址与访问地址空间对照表

数字地址范围	访　问　地　址　空　间
00H　- 7FH	直接或间接寻址的低128字节的片内RAM
80H　- FFH	间接寻址的高128字节的片内RAM
000H - 07FH	直接或间接寻址的低128字节的片内RAM

080H - 0FFH	直接寻址的SFR地址空间

四位数字地址

根据仿真模式，四位数字地址访问ME-5103或目标系统上的片外地址空间(XDATA)。见表2-3。

表2-3 四位数字地址与访问地址空间对照表

仿真模式	数字地址范围	访 问 地 址 空 间
0 和 1	0000H - BFFFH	ME-5103仿真器的存贮器空间
0 和 1	CE80H - CE9FH	ME-5103仿真器的用户参数区
0 和 1	E000H - FFFFH	ME-5103仿真器的扩展存贮器空间
2 和 3	0000H - FFFFH	目标系统的外部数据空间

NEXT,LAST -地址增量和减量键

NEXT,LAST键具有两个功能。其一：检查一个地址内的数据,其二：在显示数据时,用作地址的增量和减量。下面是 NEXT,LAST键与数字键在仿真模式0情况下的应用实例：

- 按RESET键
  ME-5103显示P
- 输入0214
  ME-5103显示0214
- 输入56
  ME-5103显示1456。如继续输入显示将滚动左移。
- 按NEXT键,检查1456H地址内的数据
  ME-5103显示1456 XX. XX为外部RAM地址1456H内的数据。
- 输入34

ME-5103显示1456 34。外部RAM地址1456H内的数据被写进新输入的数据34H。

- 输入AB
  ME-5103显示1456 AB。外部RAM地址1456H内的数据被写进新输入的数据ABH。
- 按MON键
  ME-5103返回到接收命令状态，显示P。
- 按6
  ME-5103显示6
- 按NEXT键,检查寄存器R6内的数据
  ME-5103显示6　　XX.XX为寄存器R6的数据。
- 按NEXT键,检查寄存器R7内的数据
  ME-5103显示7　　XX.XX为寄存器R7的数据。

错误！未定义书签。　　输入02

ME-5103显示7　　02.寄存器R7被写进新输入的数据02H。

- 按NEXT键,检查寄存器DPTR内的数据
  ME-5103显示8　XX XX.XX XX为寄存器DPH和DPL的数据。
- 按MON键
  ME-5103返回到接收命令状态,显示P。
- 输入07
  ME-5103显示07
- 按NEXT键,检查片内RAM 07H内的数据
  ME-5103显示07　　02。因为寄存器R7(寄存器区0时)的数据与片内RAM 07H相同。
- 按LAST键,检查片内RAM 06H内的数据
  ME-5103显示06　　XX.XX为片内RAM 06H的数据。
- 按MON键

ME-5103返回到接收命令状态,显示P。

- 输入089
  ME-5103显示089
- 按NEXT键,检查地址为SFR TMOD的数据
  ME-5103显示089　00
- 输入30,TMOD的内容改为30H
  ME-5103显示089　30
- 按LAST键,检查SFR TCON的数据
  ME-5103显示088　00
- 按LAST键,检查SFR PCON的数据
  ME-5103显示087　70
- 按LAST键,检查SFR空间地址为86H的数据
  ME-5103显示086　FF,地址为86H的SFR未定义。
- 按MON键
  ME-5103返回到接收命令状态,显示P。
- 输入0000
  ME-5103显示0000
- 按LAST键,检查外部RAM空间地址为0000H的数据
  ME-5103显示0000 XX
- 输入02,将外部RAM空间地址为0000H的数据改为02H
  ME-5103显示0000 02
- 按NEXT键,检查外部RAM空间地址为0001H的数据
  ME-5103显示0001 XX
- 按LAST键,检查外部RAM空间地址为0000H的数据
  ME-5103显示0000 02

TRACE -跟踪键

每按一次TRACE键执行一条指令，即跟踪运行。当第一次按下

TRACE 键后，ME-5103 显示当前的PC，操作码及累加器的内容。再次按下TRACE键后，执行一条指令。此时，ME-5103显示的是执行跟踪后的PC，操作码及累加器的内容。如从一个新的地址执行跟踪操作，则按 MON 键返回到接收命令状态，输入 4 位地址后，按下TRACE键，即将4位地址作为新的PC，再次按下TRACE键后，执行该条指令。

STEP -单步键

每按STEP键一次能执行一条指令，即单步运行。与TRACE不同的是，如当前执行的指令是调用指令，则将被调用的子程序一步运行完成。当第一次按下STEP键后，ME-5103显示当前的PC，操作码及累加器的内容。再次按下STEP键后，执行一条指令。此时, ME-5103显示的是执行单步后的PC，操作码及累加器的内容。如从一个新的地址执行单步操作，则按MON键返回到接收命令状态，输入4位地址后，按下STEP键，即将4位地址作为新的PC，再次按下STEP键后，执行该条指令。

HIST -历史键

当使用 TRACE或 STEP键进行跟踪和单步运行程序时，程序每步经过的PC记录在16步的历史缓冲器中。HIST键用于观察所记录的PC路径。

```
0000 0123      AJMP   0024H        ;jump start
0024 7840      MOV    R0,#40H      ;ram address
0026 7400      MOV    A,#00H       ;initial value
0028 7910      MOV    R1,#10H      ;count
002A F6        MOV    @R0,A        ;load value to ram
002B 04        INC    A            ;next value
002C 08        INC    R0           ;point to nextlocation
002D D9FB      DJNZ   R1,002AH     ;loop until done
```

上面是一个对片内RAM初始化的例子。程序经过16次循环操作后，片内RAM 40H-4FH的内容被赋值00H-0FH。如使用TRACE跟踪键执行，假设INC R0已被执行(即当前PC为002C)。

- MON键
  ME-5103返回到接收命令状态,显示P.
- 按HIST键
  ME-5103显示002B
- 按LAST键
  ME-5103显示002A
- 按LAST键
  ME-5103显示002D
- 按LAST键
  ME-5103显示002C
- 按MON键
  ME-5103返回到接收命令状态,显示P。

BP,BT -断点和断点次数键

BP键用于在用户程序中设置断点，并只允许设置一个断点。设置一个新断点，将自动清除上一个断点。BT键用于对已设断点设置循环次数，次数从1-256次。01为缺省值，即一次,00H为256次。

用HIST键中的一段程序说明BP键和BT键的用法。在四位地址后出现小写字母"c""表示设置断点。见下例：

- 按MON键
  ME-5103返回到接收命令状态,显示P。
- 输入002A
  ME-5103显示002A
- 按BP键一次
  ME-5103显示002A,并在002A的地址上设置断点。
- 再按BP键一次

ME-5103显示002A,删除在002A的地址上设置的断点。

- 再按BP键一次
  ME-5103显示002A,又并在002A的地址上设置断点。
- 输入08
  ME-5103显示08
- 按BP键
  ME-5103显示08,设置断点8次。
- 按MON键
  ME-5103返回到接收命令状态,显示P。

在已经设置的断点处按BP键或在非断点处按BP键两次都可删除断点。按RESET键复位也清除断点，但SFR也被置为上电状态。

ME-5103的断点形式为软件换字断点在设置断点时必须注意:

1.软件换字断点只适用于RAM中设置断点。

2.设置断点时，从断点开始的三个字节均被替换，在程序全速运行到达断点前不得经过断点后的两个字节。

EXEC -执行程序键

EXEC键是一个全速运行程序键。按下EXEC键即从当前的程序计数器开始运行程序。如输入四位地址后按EXEC键，将把四位地址作为当前的程序计数器开始运行程序。程序在运行时，如经过断点，ME-5103将显示断点地址，操作码。

```
0000   0123     AJMP   0024H      ;jump start
0024   7840     MOV    R0,#40H    ;ram address
0026   7400     MOV    A,#00H     ;initial value
0028   7910     MOV    R1,#10H    ;count
002A   F6       MOV    @R0,A      ;load value to ram
002B   04       INC    A          ;next value
002C   08       INC    R0         ;point to nextlocation
```

```
002D   D9FB     DJNZ   R1,002AH   ;loop until done
002F   00       NOP               ;endless loop
0030   80FD     SJMP   002FH
```

这个程序经过16次循环后进入无限循环操作。假设ME-5103是在上电后仿真模式为0的状态。

- 按RESET键
  ME-5103为初始接收命令状态,显示P。
- 输入002C
  ME-5103显示002C
- 按BP键
  ME-5103显示002C,在002A的地址上设置断点。
- 输入0A
  ME-5103显示0A
- 按BT键
  ME-5103显示0A,设置断点次数10次。
- 输入0000
  ME-5103显示0000
- 按EXEC键
  ME-5103显示002C 08
- 按MON键
  ME-5103返回到接收命令状态,显示P。
- 输入40后,按NEXT键
  ME-5103显示40　　00
- 按NEXT键
  ME-5103显示41　　01
- 按NEXT键8次
  ME-5103显示49　　09
- 在按NEXT键
  ME-5103显示4A　　XX,因为经过10次断点,4AH单元没

被写。

- 按MON键
  ME-5103返回到接收命令状态,显示P。
- 输入0000
  ME-5103显示0000
- 按EXEC键一次
  ME-5103显示002C 08.到达断点,数码管上方的8个LED二进制代码的方式显示累加器ACC的内容01。
- 按EXEC键15次
  ME-5103显示002C 08.到达断点,每按一次,数码管上方8个LED的内容改变一次。
- 按MON键
  ME-5103返回到接收命令状态,显示P。

MAP -仿真模式和写模式选择键

MAP键有两个功能:

1.改变仿真器的仿真模式，即仿真程序存贮器的空间是ME-5103的存贮器还是目标系统的程序存贮器,仿真程序存贮器的空间是ME-5103的存贮器还是目标系统的数据程序存贮器。

2.设定写模式，即EEPROM编程还是对EPROM或MCU编程。

仿真模式选择

ME-5103仿真器的仿真模式决定了在运行时，PSEN和WR,RD信号是在ME-5103的内部还是输出到仿真头，到目标系统。仿真模式由七段数码管左侧的两个LED指示。仿真模式的详细说明见附录A。表2—4说明了四种仿真模式的访问空间。

写模式选择

ME-5103仿真器对EEPROM和EPROM的写入须设置编程EEP-

ROM模式和编程EPROM模式。在运行编程命令完成后,自动清除为写RAM模式。在ME-5103 仿真器上，可直接对2864和2817A进行编程。使用ME-5103仿真器的ME-EP2编程器选件,能对2732/64 /128 /256 编程。使用ME-5103仿真器的ME-EP3编程器选件，能对8751/52,87C51/52,89C51/52编程。

表2—4 仿真模式的访问空间

仿真模式	目标程序空间	目标数据空间	访 问 空 间
0	不亮	不亮	ME-5103的程序和数据空间
1	亮	不亮	目标系统的程序空间,ME-5103的数据空间
2	不亮	亮	ME-5103的程序空间,目标系统的数据空间
3	亮	亮	目标系统的程序和数据空间

设置编程EEPROM模式

- 按MON键
  ME-5103返回到接收命令状态,显示P。
- 输入0
  ME-5103显示0
- 按MAP键
  ME-5103显示EEP,表示已经设置临时模式。

设置编程EPROM模式

- 按MON键
  ME-5103返回到接收命令状态,显示P。

- 输入1
  ME-5103显示1
- 按MAP键
  ME-5103显示EP,表示已经设置临时模式。

关于对RAM,EEPROM和EPROM的写入,见数据块命令部分。

数据块命令

ME-5103提供了一组关于对数据块操作的命令。包括数据块比较、数据块移动、数据块填充、搜寻匹配与不匹配数据等功能。在数据块比较，数据块移动时，仿真模式指示器(TARGET PROG)指示源数据块所处的空间，仿真模式指示器(TARGET DATA)指示目标数据块所处的空间。在数据块填充，搜寻匹配与不匹配数据时，仿真模式指示器(TARGET DATA)指示目标数据块所处的空间。

BTOP,BEND -设置块首和块末键

ME-5103在执行所有块命令前都必须定义一个起始地址和结束地址所包含的数据块。这个数据块的格式是：

[XXXX] <BTOP> [YYYY] <BEND>

其中:XXXX是数据块的起始地址，YYYY是数据块的结束地址。

在执行数据块命令返回监控"P"状态后，定义的数据块起始地址和结束地址保持不变，数据块标志自动清除。数据块起始地址和结束地址能够用 BTOP和 BEND键调出显示。

COMP -数据块比较键

数据块比较命令是将源数据块(由TARGET PROG指示的空间)和目标数据块 (由 TARGET DATA指示的空间) 内容进行比较的命令。对于定位在ME-5103和目标系统内不同空间的数据块比较，由仿真模式决定。见表2-5。

数据块比较命令的格式:

[XXXX] <BTOP> [YYYY] <BEND> [ZZZZ] <COMP>

其中:XXXX是源数据块的起始地址,YYYY是源数据块的结束地址,ZZZZ是目标数据块的起始地址。

表2—5

仿真模式	比　　较　　操　　作
0	比较源数据块和目标数据块都在ME-5103仿真器的数据块
1	比较源数据块在目标系统程序空间和目标数据块在ME-5103仿真器的数据块
2	比较源数据块在ME-5103仿真器和目标数据块在目标系统数据空间的数据块
3	比较源数据块在目标系统程序空间和目标数据块在目标系统数据空间的数据块

执行数据块比较命令时，将源数据块和目标数据块的数据进行字节比较。比较不等时，显示目标数据块当前的数据和地址。按BTOP键则显示源数据块的数据，按 BEND键则返回到显示目标数据块的数据。按COMP键继续比较。

MOVE -数据块移动键

数据块移动命令是将源数据块(由TARGET PROG指示的空间)的数据移动到目标数据块 (由TARGET DATA指示的空间) 中的命令。对于定位在ME-5103和目标系统内不同空间的数据块移动，由仿真模式决定。见下表:

表2—6

仿真模式	移　　动　　操　　作
0	ME-5103仿真器存贮器空间的数据块移动
1	目标系统程序空间的数据块移动到ME-5103仿真器存贮器空间
2	ME-5103仿真器存贮器空间的数据块移动到目标系统数据空间
3	目标系统程序空间的数据块移动到目标系统数据空间

数据块移动命令的格式:

[XXXX] <BTOP> [YYYY] <BEND> [0/1 MAP] [ZZZZ] <MOVE>

其中:XXXX是源数据块的起始地址,YYYY是源数据块的结束地址，ZZZZ是目标数据块的起始地址。

执行数据块移动命令时， 将源数据块的数据移动到目标数据块。如数据块移动的目标数据块写入错误将显示目标数据块写入错误时的地址和数据。当目标数据块是EEPROM或EPROM时,必须在按MOVE键前设置写模式。下面是一个对EEPROM写入的例子:

- 按MON键
  ME-5103返回到接收命令状态,显示P.
- 输入0000
  ME-5103显示0000
- 按BTOP键
  ME-5103显示0000 c
- 输入0000
  ME-5103显示0FFF

错误！未定义书签。 按BEND键
ME-5103显示0FFF c

- 输入0

ME-5103显示0

- 按MAP键
  ME-5103显示EEP
- 输入A000
  ME-5103显示A000
- 按MOVE键
  ME-5103对EEPROM进行编程后,显示P

FFIS,FFNO -搜寻匹配与不匹配数据键

搜寻匹配与不匹配数据命令是在目标数据块(由TARGET DATA指示的空间)内搜寻与输入的关键字数据匹配与不匹配的命令。对于定位在ME-5103和目标系统内不同空间的搜寻操作，由仿真模式决定。见下表:

表2—7

仿真模式	搜　　寻　　操　　作
0	ME-5103仿真器存贮器空间搜寻
1	在ME-5103仿真器存贮器空间搜寻
2	在目标系内的数据空间搜寻
3	在目标系内的数据空间搜寻

搜寻匹配与不匹配数据命令的格式:

[XXXX] <BTOP> [YYYY] <BEND> [ZZ] <COMP>

其中:XXXX是源数据块的起始地址,YYYY是源数据块的结束地址,ZZ是搜寻匹配与不匹配的关键字数据。

执行搜寻匹配与不匹配命令时，ME-5103显示在目标数据块内的第一个匹配(按FFIS键)和不匹配(按FFNO键)的数据和地址。继续按

FFIS和FFNO键，将搜寻下一个匹配与不匹配的数据。

系统实用命令

ME-5103仿真器的系统实用命令有计算相对转移的偏移量，十六进制和十进制转换，加减运算，用户定义命令和功能键等。

OFFSET -偏移量计算键

偏移量计算命令是计算相对转移指令偏移量的命令。对于计算相对转移指令偏移量，用户必须给出当前指令的地址，转移地址和转移指令的操作码。

偏移量计算命令的格式:

[XXXX] <BTOP> [YYYY] <BEND> <OFFSET>

其中:XXXX是指令操作码地址,YYYY是转移的目标地址。

在执行偏移量计算后，正确计算的偏移量自动填入指令的相对应的操作数内。对于前面的程序实例中在地址 002DH 处的DJNZ R1,002AH汇编语句，偏移量计算见以下操作:

- 按MON键
  ME-5103返回到接收命令状态,显示P。
- 输入002D
  ME-5103显示002D
- 按NEXT键
  ME-5103显示002D XX
- 输入D9
  ME-5103显示002D D9,DJNZ R1,002AH的操作码。
- 按MON键
  ME-5103返回到接收命令状态,显示P。
- 输入002D

ME-5103显示002D

- 按BTOP键
  ME-5103显示002D c
- 输入002A
  ME-5103显示002A,转移的目标地址。
- 按BEND键
  ME-5103显示002A
- 按OFFEST键
  ME-5103显示002E FB,相对转移指令的偏移量。

HEX,DEC,+,-十六进制和十进制转换,加减运算键

ME-5103仿真器提供了简单的数制转换，加减运算功能。HEX和DEC键为十六进制和十进制转换键，+和-为加减运算键。数制转换，加减运算操作见下例:

- 按MON键
  ME-5103返回到接收命令状态,显示P。
- 按HEX键
  ME-5103显示H
- 输入1FA
  ME-5103显示H 1FA
- 按DEC键
  ME-5103显示D 506
- 按+键
  ME-5103显示不改变
- 输入1274
  ME-5103显示1274
- 按+键
  ME-5103显示1780
- 再按+键
  ME-5103显示3560

- 按HEX键
  ME-5103显示H　DE8

FUNC -用户功能键

ME-5103除了提供监控程序定义的功能键外，在监控程序支持下为设计使用者附加了16个用户功能键。用户功能键所执行的程序由监控程序调用，必须用返回指令结束。用户程序的地址表存放在以0CE86H-0CE87H的为首址的32字节表中。

用户功能键的格式为:

<0-F> [FUNC]

在使用用户功能键时必须除始化用户程序的地址表的首地址和用户程序的入口地址表的内容。

用户程序的入口地址=（0-F）X 2+用户程序的地址表首址

USER -用户命令键

ME-5103除了提供监控程序定义的功能键外，在监控程序支持下为设计使用者附加了16个用户命令键。用户命令程序由监控程序转入执行，必须用转移指令转到监控程序D000H处结束。用户程序的地址表存放在以0CE84H-0CE85H的为首址的32字节表中。

用户功能键的格式为:

<0-F> [USER]

在使用用户命令键时必须初始化用户程序的入口地址表的首地址和用户程序的地址表的内容。

用户程序的入口地址=（0-F）X2+用户程序的地址表首址

ME-5103仿真器的硬件结构

由于使用了PLA (PAL, GAL)器件，对于标准含8K RAM存贮器的ME-5103仿真器，只用了11只集成电路IC。除了通用集成电路具有的功能外，U7为地址逻辑，U8为仿真控制和模式切换逻辑，U9为仿真输

出控制。在U5 (2000H-9FFFH)，U4(A000H-BFFFH)插上32K 和 8K 的RAM，可将 ME-5103仿真器的存贮器扩充到 48K。

标有JP1-JP5的跳线器，用于对ME-5103仿真器的硬件配置。 JP1选择仿真器振荡器和目标系统振荡器，2-3 相连使用仿真器振荡器，1-2相连使用目标系统振荡器。JP2将MCU片内P3.0、P3.1连至U13(MAX232)后转换为RS232电平从CN6输入输出，1-2相连引至仿真插座CN1，2-3相连引至U13。JP3 短路时，将ME-5103电源与目标系统电源连接。JP4为存贮空间A000H-BFFFH (U4)的写允许选择，开路时不允许写操作。JP5为仿真器和单板机选择，开路时为仿真器。

ME-5103仿真器有 CN1-CN6共六个连接器。CN1为仿真电缆连接插座，在仿真目标系统时，将仿真电缆的IDC40插头与CN1相连，另一头DIP40与目标MCU相连。CN2,CN3为扩展插座，用于连接ME-5103仿真器选件，如编程器 ME-EP2，ME-EP3或用户自制扩展板。 CN4, CN5为单板机方式时，分割ME-5103为双板结构的连接插座。 CN6为ME-5103仿真器与IBM PC的连接插座，插座内定义了电源引脚。关于CN1-CN6的引脚定义见附录C。

ME-5103仿真器地址空间和存贮器扩展

ME-5103仿真器提供48K(0000H-0BFFFH)的存贮器空间，作为用户的程序区和数据区。在从0000H-1FFFH的8K地址空间配有带后备电池的RAM。ME-5103仿真器的监控程序占用8K(C000H-DFFFH)的程序和数据空间。最后的8K(E000H-FFFFH)空间由CN2, CN3扩展。ME-5103仿真器地址空间和支持器件见表3-1。仿真目标系统时，用户可用程序空间为0000H-BFFFH的48K，数据空间0000H-FFFFH共64K。

表3-1 ME-5103仿真器地址空间和支持器件

地 址 范 围	长度	支 持 器 件	功 能
0000-1FFFH	8K	6264	仿真存贮器,后备电池
2000-9FFFH	32K	62256 6264 2817A	仿真存贮器

		2864 2764 27128	
A000-BFFFH	8K	6264 2817A 2864 2716 2732 2764	仿真存贮器
C000-DFFFH	8K	2764	监控程序
E000-FFFFH	8K		扩展存贮器

ME-5103仿真器用户参数区

ME5103仿真器在监控数据空间内为用户保留了地址为CE80H-CE9FH的32字节用户参数区，存放在8155片内。这些数据能被用户通过键盘或程序设置和修改。前10个字节，监控程序已定义了功能，后22个字节由用户定义使用。ME5103仿真器用户参数区在上电复位时，由监控程序初始化，初始化值和功能见表 3-2。按RESET键复位，不改变用户参数区的内容。

地　　址	初始化值	功　　能
CE80H-CE81H	DF80H	显示字符表首地址
CE82H-CE83H	DF80H	键值表首地址
CE84H-CE85H	DF80H	用户命令程序入口表首地址
CE86H-CE87H	DF80H	用户功能程序入口表首地址
CE88H	E0H	8个LED显示数据的地址
CE89H	FFH	数码管显示数据的地址
CE8AH-CE9FH	DF80H	用户使用

ME-5103显示字符表首地址(CE80H-CE81H)

上电初始化为DF80H。当用户调用监控程序中的显示子程序时,显示的字形与监控相同。如用户需自行定义显示字形，则修改CE80H-CE81H的值,指向用户的显示字形表首地址。高位为CE80H,低位为CE81H。关于调用监控程序,请见附录E.

ME-5103键值表首地址(CE82H-CE83H)

上电初始化为DFA0H。当用户调用监控程序中的取键值子程序时，所取的键值与监控相同。如用户需自行定义键值,则修改CE82H-CE83H的值,指向用户的键值表首地址。高位为CE82H，低位为CE83H。关于调用监控程序,请见附录E。

用户命令程序入口表首地址(CE84H-CE85H)

使用用户命令键，必须对CE84H-CE85H初始化。初始化后的值指向用户命令程序入口表的首地址。执行完用户程序后，使用LJMP 0D000H指令返回监控程序。高位为CE84H,低位为CE85H。

用户功能程序入口表首地址(CE86H-CE87H)

使用用户功能键，必须对CE86H-CE87H初始化。初始化后的值指向用户功能程序入口表的首地址。执行完用户程序后，使用RET指令返回监控程序。高位为CE86H,低位为CE87H。

8个LED显示数据的地址(CE88H)

ME-5103仿真器七段数码管上方的 8个 LED组成二进制显示器,用于显示一个二进制数。每执行一步程序,显示器显示由CE88H 定义的MCU片内存贮单元的内容，上电初始化为E0H。其值00-7F为片内RAM，80-FE为SFR，FF为程序操作码。

数码管显示数据的地址(CE89H)

ME-5103仿真器的七段数码管的低两位，在每执行一步程序时,显示由 CE89H定义的 MCU片内存贮单元的内容。上电初始化为FF。其值00-7F为片内RAM,80-FE为SFR，FF为程序操作码。

ME-5103单板机

短路JP5即可将ME-5103仿真器作为单板机使用。在作为单板机使用时，8155(U10)的I/O口，RAM均可被用户使用。8155的地址分配见

表3-2。

表3-2　8155的地址分配

地　　址	功　　能
CE00H-CEFFH	RAM
CF00H	口控制寄存器
CF01H	口A
CF02H	口B
CF03H	口C

中断与定时器的处理

中断服务优先级寄存单元ISP

ME-5103仿真器监控程序设置了ISP寄存器(使用地址F9H)。在实际运行程序时，是不访问这个寄存单元的。它只用于监控程序正确地对中断进行处理，保证用户程序的运行。ISP可用访问SFR的方式检查和修改。

在执行跟踪、单步运行用户程序时，监控程序把当前的中断状态自动记录到 ISP的相应位，使监控程序了解正在执行的程序是否是中断程序及高低优先级。如全速执行用户程序到达断点，监控程序就无法判断当前的中断状态，即当前程序是否是中断程序。

中断服务优先级寄存单元ISP的使用规则

1.跟踪，单步运行用户程序时，监控程序自动判断当前的中断状态，无需对中断服务优先级寄存单元处理。即使用跟踪，单步进入和退出中断，用户不必处理ISP。

2.全速运行用户程序到达断点并返回监控程序时，如当前程序是中断程序，则根据中断的优先级设置ISP的高半字节或低半字节。

3.全速运行用户程序到达断点并返回监控程序时，如当前程序不是中断程序，则根据中断的优先级清除ISP的高半字节或低半字节。

当程序响应中断时，ISP指示当前的中断优先级和中断源，ISP的高半字节指示低优先级中断源，低半字节指示高优先级中断源，按MCU的查询次序排列：

表3-3

ISP值	标　号	中　　断　　源
0	无	
1	X0	外部中断　　INT0
2	S1	串行通讯　　SIO1
3	ADC	模数转换　　ADC
4	T0	定时器　　T0
5	CT0	捕获寄存器　CT0
6	CM0	比较寄存器　CM0
7	X1	外部中断　　INT1
8	CT1	捕获寄存器　CT1
9	CM1	比较寄存器　CM1
A	T1	定时器　　T1
B	CT2	捕获寄存器　CT2
C	CM2	比较寄存器　CM2
D	S0	串行通讯　　SIO0
E	CT3	捕获寄存器　CT3
F	T2	定时器　　T2

ME-5103仿真器对关于中断指令的处理

ME-5103仿真器在跟踪，单步执行以下指令时，采用模拟的方法执行:

MOV(ORL,ANL)　　IE,A

```
MOV(ORL,ANL)    IP,A
MOV(ORL,ANL)    IE,#DATA
MOV(ORL,ANL)    IP,#DATA
PUSH            IE
PUSH            IP
POP             IE
POP             IP
MOV             IE.i
MOV             IP.i
SETB(CLR,CPL)   IE.i
SETB(CLR,CPL)   IP.i
RETI
```

其它对IE,IP操作指令,必须用断点的方法调试。

ME-5103仿真器对定时器控制

当用户程序中使用了定时器，并使其工作在定时器方式(C/Tx=0,且定时器1不作波特率发生器)时，监控程序在运行用户程序前，根据用户设定的TRx状态启动或关闭定时器，当返回到监控程序后，无论定时器是否运行，都将关闭定时器，以使用户静态地观察定时器的工作。

由于监控程序在运行用户程序前，启动定时器，返回监控程序后，关闭定时器，会在定时器上附加计数。ME-5103仿真器对这附加计数不去处理，根据需要让用户修正。如ME-5103仿真器处理附加计数，可能会引起错误，反而不利调试。表 3-3为监控程序处理定时器时附加计数所需的T状态数。

表3-4.监控程序处理定时器时附加计数所需的T状态数

	监控到用户程序	用户到监控程序	监控到用户到监控程序
定时器 0	4T	28T	32T
定时器 1	4T	28T	32T

定时器 2	6T	32T	38T

第二部分 MBUG调试器

MBUG调试器是万利电子有限公司为ME-5103单片机仿真开发系统编制的仿真调试软件，可在IBM PC及兼容机上运行。

使用MBUG软件调试器及交叉汇编器，能够对用户的源程序进行汇编，装入目标代码到ME-5103单片机在线仿真器或目标系统，仿真用户目标系统的软、硬件，具有跟踪、单步、断点设置、全速运行等多种调试命令。

当ME-5103 仿真器处于"P"状态时，MBUG采用随机方式与ME-5103仿真器串行通讯，波特率自适应。在非通讯状态期间，ME-5103仿真器和系统机都可以分别进行操作。

MBUG将屏幕划分为寄存器窗口、存贮器窗口、工作窗口、标号窗口、变量窗口、文件名窗口及主菜单命令等多个窗口。ME-5103仿真器的状态、命令提示、调试过程中的出错信息显示在屏幕的顶行和底行，用户无须切换即可得到各种调试信息。

支持对用户ASM、PL/M51及C-51语言源文件的文本、标号、反汇编方式调试。在线行汇编、数据修改，单键触发并带有提示帮助的热键和命令键，内含交叉汇编等丰富的调试运行功能，提高用户仿真调试效率。

安装和运行MBUG

MBUG调试软件存放在一张软磁盘上。安装时，将文件名含有MBUG的文件拷贝到用户当前目录或DOS路径能访问到的目录下，键入MBUG进入调试器运行状态。进入MBUG后，按任意键即可工作。

MBUG窗口环境

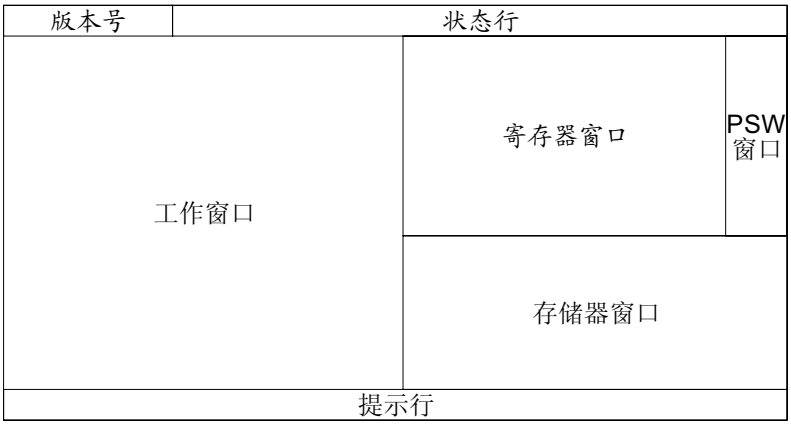


图2-1 MBUG的窗口

状态行

MBUG在屏幕的顶行显示：MBUG版本号、仿真MCU类型、装入调试程序的文件名、仿真模式和串行口信息等。

工作窗口

MBUG的工作窗口在装入程序后，显示被调试程序的源文件文本、标号或反汇编代码；在EPROM编程操作时，用于输入命令；反汇编时，显示已反汇编的代码、助记符。

寄存器窗口

显示特殊功能寄存器SFR、寄存器R0-R7或片内变量。此窗口的显示内容能被用户定义。

存贮器窗口

显示用户指定存贮器区域的内容。

命令和错误指示信息行

命令和错误指示信息行位于屏幕的底行，显示当前可操作的命令信息和发生的错误。

变量表及标号表窗口

这是一个弹出式窗口。出现在RUN模块中，显示变量、标号的地址和属性。

对话和行汇编窗口

MBUG的大多数命令为单键输入，但有时需要输入一串字符，例如输入地址、标号、文件名或汇编语句等，用户可以在此窗口下输入。

变量窗口

变量窗口与寄存器窗口交替显示。变量窗口是按用户程序中定义的变量、标号的先后次序，显示当前的内容。

汇编窗口

MBUG调试器内含的交叉汇编器弹出式窗口对话窗口。

常用的功能键

上下左右光标键	-上下移动一行或移动一列
PgUp,PgDn	-上下翻动一页
Ctrl-PgUp,Ctrl-PgDn	-移动光标到文件开始或结束
HomeEnd	-移动光标行开始或结束
Ctrl-Home,Ctrl-End	-移动光标窗口顶行或底行
Tab	-切换激活窗口

MBUG主菜单命令

选择主菜单命令可以用下面几种方法：

1.使用光标键移动光带，按回车键。
2.直接按主命令首字母键。
3.按Alt键和加亮字符。

Help，？ 帮助

MBUG的帮助功能只显示 MBUG.HLP文件。 由 F1键触发帮助，屏幕上显示帮助文件的内容。

OS shell DOS功能调用

DOS功能调用。进入操作系统并在操作系统提示符下，用户可以执行操作系统的各种命令。在DOS提示符下，键入EXIT命令返回MBUG。

Load 装入调试文件

装入格式为INTEL HEX或INTEL OMF的文件。由状态行I_PS或U_PS所指示的方向，将代码文件装载到 ME-5103仿真器的存贮器或用户目标系统的存贮器。如果目标文件是由MBUG的汇编器汇编的，装入过程自动在代码文件中获得有关MCU信息，改变当前仿真目标MCU的类型。

MBUG对被调试文件的文件名输入作如下规定:

1.用MBUG汇编的产生的目标文件，直接输入文件名(扩展名必须为.ASM)或扩展名为.HEX的文件。

2.用INTEL ASM51加DEBUG调试项汇编，经RL51连接的汇编程序，输入源程序文件名，扩展名必须为.ASM。

3.用INTEL PLM51加DEBUG调试项编译，RL51连接的PLM51语言程序，应输入源程序文件名，扩展名必须为.PLM。

4.符合INTEL OMF标准的C语言程序，应输入源程序文件名,扩展名必须为.C。

5.用Archimedes C-51语言编程的程序，经编译连接通过后，输入源程序文件名，扩展名为.C。

6.源文件文本调试为单模块调试。

Setup 设置寄存器窗口

MBUG支持用户修改寄存器窗口的显示格式和内容，使被调试程序中所使用的MCU片内存贮器变量在窗口内显示。

进入SETUP模块，用户即可编辑这个窗口，编辑显示格式定义如下:

<变量名>=<地址><类型符>

其中:	变量名	-定义片内直接访问数据存储器的符号
		-定义可位寻址的符号
		-SFR特殊功能寄存器
	地址	-二位16进制数,不要加前导零
	类型符	-H或. 16进制方式显示
		-Q或% 二进制方式显示
		-空格 位类型显示

编辑完成后使用以下命令:

F2	SAVE&UPDATA	编辑写盘并更新
F3	UPDATA	更新
F4	RESTORE	放弃当前编辑的内容,MBUG.CFG重新装入数据
F10	Menu	返回主菜单

Assemble 交叉汇编器

MBUG的交叉汇编器。将汇编语言源程序汇编成INTEL HEX代码文件，见交叉汇编部分。

Disassemble 反汇编器

MBUG反汇编器.将INTEL HEX代码文件反汇编成汇编语言助记符文件。

RUN 运行

调试运行程序。以下是调试运行程序的命令:

Ctrl-S	选择仿真模式
Ctrl-C	选择串行口
Ctrl-E	选择MCU
Ctrl-O	更新MBUG屏幕
Ctrl-B	设置断点
Ctrl-T	设置断点重复次数
Ctrl-N	设置程序计数器PC
Ctrl-D	改变显示格式(工作窗口)
Ctrl-M	改变存贮器显示空间
Ctrl-G	指定窗口显示首地址
F2: BP	设置断点
F3: PC	设置程序计数器PC
F4: Here	执行程序到光带处
F5: Scal	单步
Ctrl-F5:	自动单步
F6: ZOOM	改变窗口尺寸
Ctrl-F6:	窗口交换
F7: Symbol	改变显示格式(工作窗口)切换寄存器/变量窗口
Ctrl-F7:	激活标号窗口
F8: Step	跟踪
Ctrl-F8:	自动跟踪
F9: Exec	全速运行
F10: Menu	返回主菜单

Quit 退出 MBUG

Alt-X　　返回 DOS 操作系统。

调试运行

这是 MBUG调试器的一个主要模块。用户在仿真调试过程中，运行程序、修改数据、观察变量都可以在模块内完成，无须记忆命令，简单方便。

MBUG具有标号、符号、源文件文本调试功能。在进入MBUG后首先将调试的目标代码用LOAD装入ME-5103仿真器或用户系统,此时在工作区窗口上将显示用户源程序文本，然后即可进入调试运行过程。装入的代码文件，必须是用MBUG中Assemble汇编器汇编产生的代码文件，或是用INTEL ASM51/PLM51带 DEBUG 控制项汇编编译产生的目标文件。通常在LOAD 模块装入代码文件成功之后， 直接进入RUN调试运行模块运行。

装入目标代码后按任意键进入RUN模块，MBUG通过串行口装入ME-5103仿真器当前状态,刷新显示寄存器，存贮器窗口和工作窗口并从当前PC地址开始显示程序。

工作窗口显示用户程序，并以三种方式显示：源文件文本方式,标号方式和代码方式。 存贮器窗口显示当前选定的存贮空间的数据。寄存器窗口显示寄存器R0-R7,SFR的内容。

调试运行命令

F2 BP,Ctrl+B　设置断点 ／ Ctrl+T BT　设置断点次数

-使用光标键移动光带到要设置的断点处按F2键。

-设置的断点地址不在当前窗口内,用Ctrl+B命令设置。

-在标号窗口内直接设置断点。Ctrl+F7键激活标号窗口，移动

光带到目标标号上，按F2键设置断点后，断点处的语句行用红色光带指示，按Ctrl+T键输入断点循环次数。

ME-5103的断点形式为软件换字断点。在设置断点时必须注意:

1.软件换字断点只适用于RAM中设置断点。

2.设置断点时，从断点开始的三个字节均被替换，在程序全速运行到达断点前不得经过断点后的两个字节。

下面的例子说明断点位置的正确设置:

例一:

```
RETRY:      ACALL   SUBJECT
            JC      EXIT
            SJMP    RETRY       ;(1)
SUBJECT:    ADD      A, #5
            ......
            RET
```

例二:

```
            LCALL   WAT_KB
            MOV     KB,A
            ACALL   UDELAY
            RET                 ;(2)
UDELAY:     MOV     R7,#0
            DJNZ    R7,$
            RET
```

(1)如果在此设置断点，SJMP起三个字节被替换，因SJMP是二字节指令，在执行到SUBJECT子程序时，第一个指令被替换， 引起出错。

(2)此处也不能设断点。因为此处RET以及标号UDELAY处二

字节指令被替换，程序不能正常执行。

这二个例子的其余指令处地址均可设置断点。

F3 PC,Ctrl+N NewPC 设置程序计数器

移动光标键使光带停留在新的指令行处，按Ctrl+N键，光带处的地址作为新的PC被设置；若使用F3键，MBUG要求输入新的PC值(可输入十六进制数或标号)。

F4 Here 运行程序到光带处

在移动光带浏览程序时，按F4键使程序全速运行到光带处。这个功能使用户不要通过设置断点的方式，自动设置断点并运行程序。在执行Here命令期间，用户先已设置的断点暂时被屏蔽，待程序执行到达"这里"后，恢复已设置的断点。

F5 STEP,Ctrl+F5 单步, 自动单步

每按 F5 键一次单步执行一条指令。当执行的指令是调用指令时，则将被调用的子程序一步全速运行完成。这对已调试完的功能模块的调试是非常实用的。这要求用户在编写程序时，应注意结构化,模块化设计，不仅使程序的易读，更有助于调试方便。单步在处理非调用指令时，与跟踪命令一样。在单步命令执行调用程序期间,不响应断点。

Ctrl+F5为自动单步运行。

F6 Zoom 改变窗口尺寸

按F6键改变当前已激活的工作窗口、寄存器窗口和存贮器窗口的尺寸。

F7 Symbol 标号开关

工作窗口有三种方式： 按F7键可以使工作区窗口的显示方式在标

号或非标号方式或源文件方式中切换。

-在非标号方式下，窗口显示16进制地址、程序代码及反汇编程序清单。

-在标号方式下，窗口显示程序标号，用户程序语句。

-源文件文本方式下，程序直观，方便高级语言调试。

当光带进入寄存器窗口时， F7键交替显示寄存器窗口和变量窗口。

Ctrl+F7 激活标号表窗口

按 Ctrl+F7键激活弹出式标号窗口，显示所有的标号或符号。在激活标号表窗口下支持Ctrl+G、Ctrl+N、F2等键操作。

F8 Trace,Ctrl+F8 跟踪, 自动跟踪

跟踪执行键。每按一次F8键，与ME-5103仿真器一样执行一条跟踪命令。按Ctrl+F8键，自动连续执行跟踪，按任意键停止。

F9 Exec 全速运行

全速运行用户程序。

F10 Menu 返回主菜单

返回主菜单。

数据块操作命令

数据块操作命令中定义的数据块由仿真模式决定。参见ME-5103仿真器中数据块操作命令。

Shift+F2 Comp 比较

源数据块与目标数据块比较。比较不同时，屏幕显示数据不同处的地址与数据。

Shift+F3 Move 移动

源数据块的数据复制到目标地址处。包括ME-5103仿真器与用户系统之间相互传送。

Shift+F4 Fill 填充

数据块填充命令。在指定数据地址范围内填充数据。

Shift+F5 Match 寻找相同关键字节

在指定数据块范围内寻找与关键字节相同的数据地址。找到时,显示地址及数据。

Shift+F6 Unmatch 寻找不同关键字节

在指定数据块范围内寻找与关键字节不相同的数据地址。找到时，显示地址及数据。

Shift+F7 Save 保存程序代码到磁盘文件

将 ME-5103仿真器仿真器存贮器或用户目标系统中程序空间的数据以INTEL.HEX文件格式存入磁盘，并把当前目标MCU信息也写到文件中。读取代码空间由状态行I_PS或U_PS指示。

行汇编功能

MBUG为用户提供了直接用汇编语言修改程序的功能。移动光标键使光带移到欲修改的地址处，输入汇编语句，MBUG立即将这一行语句汇编成目标代码，以覆盖方式送入ME-5103仿真器程序区或用户系统。

第三部分 MBUG汇编器

MBUG汇编器作为MBUG调试器中的一个独立模块，在IBM PC或兼容机上对用户的源程序进行交叉汇编后，产生扩展名为HEX的目标代码文件，提供MBUG调试器使用。

MBUG汇编器的字符

下列字符为有效字符:

-数字: 0-9、字母A-Z,a-z 字母不分大小写
-分隔符:冒号,空格,制表符,逗号,单引号和双引号
-字符"$"
-分号
-下划线

MBUG汇编器的符号

符号包括标号、寄存器名、替代名及指令名等。符号必须由字母开始。

MBUG汇编器的常数

常数可为二进制、十进制和十六进制常数,其格式为:

-二进制数必须以字母B结尾。如00001111B。
-十进制数以字母D结尾.缺省时为10进制数.如869D, 1265。
-十六进制数以字母H结尾。若第一个字符为字母。则必须有前导零。如1FH, 0FFFFH。
-数值可以是有符号数。 如-1, +28。
-ASCII 字符串。如'Y', 'Yes'。

MBUG汇编器对文件的限制

-源文件长度不得超过10000行。

-源程序的行长不得超过128个字符。
-符号长度不超过16个字节。
-符号不得超过3000个(长度为8个字节时)。

MBUG汇编器的运算符+,-,HIGH,LOW:

-运算符"+","-"在表达式中作正负号或作加减运算符。
-HIGH取字表达式值的高位字节。
-LOW取字表达式值的低位字节。

例:

```
DEL_VAL     EQU     103AH
                    MOV         A,#0
DELAY:              LCALL       SUB1
                    CJNE        A,#5,$-3
;                   .....
SUB1:               MOV
            R6,#HIGH(DEL_VAL)
                    MOV         R7,#LOW(DEL_VAL)
                    DJNZ        R7,$
                    DJNZ        R6,$-4
                    INC         A
                    RET
```

MBUG汇编器的伪指令

DB

用途: 字节类型代码存储器赋值
格式: DB 表达式
说明: 表达式可以是下列的一种形式:
-无符号数。数值的范围 0-255。
-有符号数。数值的范围 -128 - +127。
-符号名。

-常数表。各元素用逗号分隔。

-字符串。字符串用单引号包围,如果单引号做为字符串的一部分,则字符串必须用双引包围。

例:

```
X                       EQU     5AH
CK_FLAG         DATA    21H
STR1                    EQU     'ABC'
                        ORG     100H
TABL:                   DB      1,2,X,STR1,"Can't do",-25,+100
                        DB      DPH,DPL,TR0,CY,PSW.1,CK_FLAG.3
;                       ......
```

从100H单元开始依次存放的数为1,2,5AH,41H,42H,43H,... 。

DW

用途: 字类型代码存储器赋值

格式: DW 表达式

说明: 表达式可以是下列的一种形式:

-无符号数。其值的范围0 - 65535。

-有符号数。其值的范围-32768 - +32767。

-程序中的标号。

-常数表。元素使用逗号分隔。

例:

```
ORG     100H
DISP    EQU     0DE00H      ;外部标号定义
LABEL   MOV     A,#5        ;内部标号
ORG     200H
DW      LABEL, DISP, 1234, 2A34H, 50H-300
```

从200H单元开始依次存放数 1,0,0DEH,0,4,0D2H,2AH,... 。

DS

用途: 保留代码存贮空间

格式: DS 常数

说明: 保留由常数给定长度的代码存贮空间,并赋值零。

例:

```
ORG       100H
 DS         128
```

从100H开始的128个字节内容均为0。

EQU

用途: 定义表达式

格式: 表达式或字符串名 EQU 表达式或字符串

说明: 表达式或字符串名必须是以字母开始。

例:

```
COUNT              EQU      10
SPACE              EQU      10H
DISP               EQU      0DE00H
ORG_ADR1     EQU      100H
START              EQU      MAIN
                        ORG      ORG_ADR1
MAIN:              MOV      R0,#COUNT
                        MOV      A,#SPACE
                        LCALL    DISP
                        MOV      DPTR,#START
```

XDATA

用途: 定义外部数据存贮空间变量

格式: 变量名 XDATA 外部数据存贮空间地址

说明: 变量名必须是以字母开始。同一单元地址可以有多个变量名,但不能重新定义。

DATA

用途: 定义片内直接寻址存贮空间变量

格式: 变量名 DATA 片内直接寻址存贮空间变量地址

说明: 变量名必须是以字母开始。同一单元地址可以有多个变量名,但不能重新定义。

例:

```
ERROR           DATA    60H
IO_PORT         XDATA   0CF01H
        ORG     100H
MAIN:           MOV     R0,ERROR
                CJNE    R0,#3,M1
M1:             JNC     TALK
;       .......
TALK:           MOV     DPTR,#IO_PORT
                LCALL   DISP
```

BIT

用途: 定义可位寻址的位变量

格式: 位变量名 BIT 位地址

说明: 位变量名必须是以字母开始。同一位地址可以有多个变量名,但不能重新定义。

例:

```
SW1         BIT     36H
SW2         BIT     37H
CONTROL     DATA    2FH
TRAC_FLAG   BIT     CONTROL.3
```

```
                ORG     100H
MAIN:           MOV     C,SW1
                MOV     C,SW2
                ORL     C,/TRAC_FLAG
```

.8XXXX

用途: 指定MCU

格式: .8XXXX

说明: 汇编控制指令。使用该语句,汇编器按指定MCU的SFR标号汇编。MCU型号有：8051,8052,80652,80552,8751。缺省时由状态行MCU 型号指定。

ORG

用途: 程序代码段定位

格式: ORG 表达式

说明: 汇编器将表达式的值装入汇编地址定位计数器。

例:

```
    LOC_TMP     EQU     8000H       ;改变LOC_TMP的值实现程序定位
                ORG     0+LOC_TMP
    ;           ......
                ORG     3+LOC_TMP
    ;           ......
                ORG     100H+LOC_TMP
    ;           ......
```

END

用途: 程序结束指令

格式: END

说明: 在汇编过程中只要出现该指令，无论后面是否还有其它指令,汇编即结束。

错误信息

错误提示信息	注　　释
Syntax error	语法错
Extra characters on line	存在多余的字符
Oprands is lost	缺少操作数
ORG address is less	地址覆盖,无效的ORG地址
Invalid operand	无效的操作数
Invalid instruction	无效的指令
Invalid data	无效的数据
Multi-defined	标号重复定义
Undefined	标号未定义
Address out of range	地址超出范围
Invalid source file	无效的源文件

附录A　ME-5103仿真器模式

ME-5103仿真器对程序和数据空间的仿真有四种仿真模式。即PSEN引脚的信号引到ME-5103仿真器存贮器或目标系统，RD/WR引脚的信号引到ME-5103仿真器存贮器或目标系统。它们组合成四种方式，分别由仿真模式指示器指示。

仿真模式0

MCU的 PSEN,RD/WR信号指向ME-5103仿真器的存贮空间。ME-5103 仿真器仿真存贮空间的程序空间和数据空间合用同一存贮器，这时取指(CODE)和输入/输出(XDATA)均为同一存贮器。通常在没有目标系统时，调试程序。

仿真模式1

MCU 的 PSEN 信号指向用户目标系统，RD/WR 信号指向 ME-5103仿真器的存贮空间。在这种模式下，在调试目标系统程序存贮器时，将ME-5103仿真器存贮空间用作仿真用户程序的外部数据区。通常调试目标系统在没有连接输入/输出设备时，测试目标程序。

仿真模式2

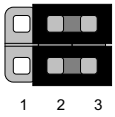
MCU的PSEN信号指向ME-5103仿真器的存贮空间，RD/WR信号指向用户目标系统。在这种模式下，在调试用户程序时，检查目标系统存贮器或输入/输出设备。这是最常用的仿真模式。

仿真模式3

MCU的PSEN,RD/WR信号都指向用户目标系统。常用于在用户程序调试完成后，检查目标系统的程序和数据存贮器或输入/输出设备是否正常运行。在这种模式下运行通过的程序，一般都能脱机运行。

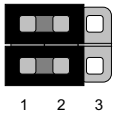
附录B　ME-5103仿真器跳线器

JP1　晶体振荡器内/外部选择



JP1

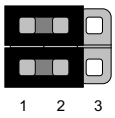
使用ME-5103振荡器



JP1

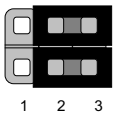
使用目标系统振荡器

JP2　MCU片内串行口TTL-RS232电平转换选择



JP2

连至仿真插座



JP2

连至MAX232

JP3　用户系统电源供电选择

JP3

ME-5103与目标系统电源断开

JP3

ME-5103与目标系统电源连接

JP4　U4(0A000H-0BFFFH)用户写允许选择

JP4

用户程序写禁止

JP4

用户程序写允许

JP5　仿真器/单板机选择

JP5

ME-5103仿真器方式

JP5

ME-5103单板机方式

附录C ME-5103仿真器连接器引脚定义

CN1 仿真插座

引脚	功能	引脚	功能
1	NC	2	P1.0
3	P0.0	4	P1.1
5	P0.1	6	P1.2
7	P0.2	8	P1.3
9	P0.3	10	P1.4
11	P0.4	12	P1.5
13	P0.5	14	P1.6
15	P0.6	16	P1.7
17	P0.7	18	RST
19	NC	20	P3.0
21	ALE	22	P3.1
23	TPSEN	24	P3.2
g 25	P2.7	26	P3.3
27	P2.6	28	P3.4
29	P2.5	30	P3.5
31	P2.4	32	TWR
33	P2.3	34	TRD
35	P2.2	36	TX2
37	P2.1	38	TX1
39	P2.0	40	GND

CN2,CN3 扩展插座

引脚	功能	引脚	功能
1	GND	2	GND
3	Vcc	4	Vcc
5	INT0	6	WR
7	RST	8	RD
9	PSEN	10	ALE
11	A8	12	AD0
13	A9	14	AD1
15	A10	16	AD2
17	A11	18	AD3
19	A12	20	AD4
21	A13	22	AD5
23	A14	24	AD6
25	A15	26	AD7

附录D ME-5103 通讯电缆连接

通讯电缆连接

ME-5103 信 号	ME-5103 9芯D型插座		IBM PC 25芯D型插座	IBM PC 9芯D型插座
+5V	1			
+5V	2			
RST	3			
GND	4			
GND	5	---	7	5
TXD	6	---	3	2
RXD	7	---	2	3
TXD(P3.1)	8			
RXD(P3.0)	9			

附录E ME-5103 监控子程序

```
CHAR_BUF        equ     3FH
PORT_A          equ     0CF01H
PORT_B          equ     0CF02H
PORT_C          equ     0CF03H
;程序入口地址 0DE60H
;检查键数入
;调用该程序返回时, ACC不等于零为键入
;入口: 无
;使用: ACC,DPTR
;出口: ACC=0 无键 ACC=XXH 有键
;*************** Routins to Test for key *************
TEST_KB:    MOV     DPTR,#PORT_A ;pointer to 8155 port A
            CLR     A            ;no outputs selected
            MOVX    @DPTR,A      ;switch off display
            INC     DPTR         ;point to 8155 port C
            INC     DPTR
            MOVX    A,@DPTR      ;read keyboard
            CPL     A            ;active key high
            ANL     A,#0FH       ;maskk off top[ nibble
            RET

;程序入口地址0DE00H
;刷新显示数码管和8个LED
;入口:   R0指向显示缓冲区高位地址
;使用:   ACC,DPTR,R3,R6,R7
;出口:   无
```

```
;*************** LED Display Routine ****************
DISP:       MOV     R0,#CHAR_BUF ;address of character
                                 ;buffer
DISP_0:     MOV     R3,#01000000B ;character position
            MOV     A,@R0        ;get binary ACCdisplay
            SJMP    DISP_2       ;go display character
DISP_1:     MOV     DPTR,#CHAR_TAB
            MOV     A,@R0        ;get character code(offset)
            ANL     A,#80H       ;get cursor flash bit
            MOV     R6,A         ;save
            MOV     A,@R0        ;get character code(offset)
            ANL     A,#7FH       ;mask off cursor flash bit
            MOVC    A,@A+DPTR    ;get character pattern
            ORL     A,R6
            MOVC    A,@A+DPTR
DISP_2:     CPL     A            ;need reverse polarity
            MOV     DPTR,#PORT_B ;pointer to 8155 port B
            MOVX    @DPTR,A      ;output char pattern
            MOV     A,R3         ;get char display position
            DEC     DPL          ;point to 8155 port A
            MOVX    @DPTR,A      ;output position
            MOV     R7,#0        ;delay 1ms
            DJNZ    R7,$
            CLR     A            ;switch off
            MOV     DPTR,#PORT_A   ;previous character
            MOVX    @DPTR,A
            DEC     R0           ;next character
            MOV     R3,A         ;get character position
            RR      A            ;change to next position
```

```
                MOV     R3,A            ;resave character position
                JNB     ACC.7,DISP_1    ;if not finished,do next
                MOV     A,R0            ;restore R0
                ADD     A,#7
                MOV     R0,A
                RET

CHAR_TAB:       DB      3FH,06H,5BH,4FH,66H,6DH,7DH,07H
                DB      7FH,6FH,77H,7CH,39H,5EH,79H,71H
                DB      00H,73H,58H,4CH,76H,5EH


;程序入口地址0DE70H
;扫描键盘并刷新显示程序.
;入口:  无
;使用:  ACC,DPTR,R2,R4,PSW
;出口:  ACC
;************* Keyboard Scan Routins ***************
KB_SCAN:        MOV     R0,#CHAR_BUF ;address of char buffer
KB_0:           ACALL   DISP_0       ;refresh display
                ACALL   TEST_KB      ;test keyboard
                JZ      KB_0         ;jmp if no key
                ACALL   KEY_CHK      ;refresh display and delay
                JZ      KB_0         ;jmp if no key
KB_SCAN_1:      CLRf    A            ;switch off current
                CPL     A            ;selected LED display
                MOV     DPTR,#PORT_B
                MOVX    @DPTR,A      ;display off
                MOV     R2,#0FEH     ;keybd column scan position
```

```
        MOV     R4,#0           ;column scan count
KB_1:   MOVDPTR,#PORT_A
        MOV     A,R2            ;output keybd scan byte
        MOVX    @DPTR,A         ;to drive 1 keybd column
        INC     DPTR            ;point to 8155 port C
        INC     DPTR
        MOVX    A,@DPTR         ;read a key
        CPL     A               ;change positive polarity
        ANL     A,#0FH          ;mask off top nibble
        JNZ     KB_2            ;jump if key pressed
        INC     R4                 ;increment column count scan
        MOV     A,R2            ;get column scan position
        JNB     ACC.7,KB_6 ;exit if all cols scanend
        RL      A               ;else next keybd column
        MOV     R2,A            ;resave column scan position
        SJMP    KB_1            ;go scan next column
KB_2:   RRC     A               ;B0=1 - key pressed in row1
        JC      KB_3            ;then R4 comtains clo no
        XCH     A,R4            ;else R4+8 for each extra
        ADD     A,#8            ;row to give keytab offset
        XCH     A,R4            ;from 0-31
        SJMP    KB_2            ;
KB_3:   MOV     R2,#21          ;key debounce delay
KB_4:   ACALL   DISP_0          ;refresh display and
        ACALL   T_KB            ;test keyboard
        JZ      KB_5            ;key released - go look-up
        DJNZ    R2,KB_4         ;wait for key release
KB_5:   MOV     DPTR,#KEY_TAB ;pointer to key table
        MOV     A,R4            ;get the offset
```

```
                MOVC    A,@A+DPTR   ;get keyboard
KB_6:           RET

;***********  Routine to Check Key Pressed **********
KEY_CHK:        ACALL   DISP_0      ;refresh display DISP_0
                ACALL   DISP_0
                ACALL   TEST_KB     ;test keyboard
                RET
KEY_TAB:        DB      07H,08H,09H,0AH,1CH,1DH,1EH,11H
                DB      04H,05H,06H,0BH,1AH,1BH,1FH,10H
                DB      01H,02H,03H,0CH,18H,19H,14H,15H
                DB      00H,0FH,0EH,0DH,16H,17H,12H,13H
```

经销商:

万利电子有限公司
南京市珠江路603-6号
邮编:210018
Tel: (25) 336 3973
Fax: (25) 771 5058